



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/757,975	01/15/2004	Pawel Z. Chadzynski	20768/2012	4258
29934	7590	05/27/2009	EXAMINER	
PALMER & DODGE, LLP				VU, TUAN A
RICHARD B. SMITH				
111 HUNTINGTON AVENUE				
BOSTON, MA 02199				
				2193
ART UNIT		PAPER NUMBER		
MAIL DATE		DELIVERY MODE		
05/27/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/757,975	CHADZYNSKI, PAWEŁ Z.	
	Examiner	Art Unit	
	TUAN A. VU	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 3/30/09.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-7,9-12 and 17-31 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-7,9-12 and 17-31 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ . |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date <u>9/30/08,3/30/09</u> . | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 3/30/09.

As indicated in Applicant's response, claims 1, 17, 23-24, 26 have been amended, and claims 8, 13-16 canceled. Claims 1-7, 9-12, 17-31 are pending in the office action.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

3. Claims 1-7, 17-21, 23-25, 26-29, 30-31 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Claims 1, 17, 23, 24, 26 recite 'allowing the user to delay the instantiation ... until a time determined by the user' and 'allowing the user ... to manipulate a first aspect of the design before loading'. According to the Specifications (para 0038, pg 9-10), these user-drive actions are not actually enabled by the product or means possessed by the inventor, but rather performed manually or realized by human desire. The recited 'delay the instantiation' and 'manipulate a first aspect of the design' cannot be viewed as a functionality owned by the Inventor; that is, the so-recited method (cl. 1, 26), or computer medium/product (cl. 17, 23, 24) when deployed or executed cannot be construed as actually carrying out the action of 'delaying' or 'manipulating' per se. The Specifications is silent about details that would explicitly put forth how user's delaying and user's manipulation are realized via a runtime software produced by the Inventor,

realization not by a human but by means provided by the invention. The above limitations are not given patentable weight since they are not enabled by means possessed by the inventor, because one of ordinary skill in the art cannot make use of those actions that strictly depends on another human desire and mood. The user-driven “delaying” and “manipulating” actions will be interpreted without weight and the loading (of *application state file* from others) as claimed will be broadly treated as having independency with respect to the user ongoing application.

Further, **claim 30** recites ‘allow the user to refuse’ and for the same reasons, this ‘refuse’ limitation will be treated broadly without patentable merits.

Dependent claims 2-7, 18-21, 25, 27-29, 30-31 fail to provide teaching as to how user actions are actually implemented by the method or product recited in the independent claims; hence are likewise rejected for lack of enablement support.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-7, 9-12, 17-31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Notani et al, USPN: 6,567,783 (hereinafter Notani) and further in view of Thackston, USPN: 6,928,396 (hereinafter Thackston).

As per claim 1, Notani discloses a computerized method for collaborating over a network (col. 1 li. 50 to col. 2 line 24; Fig. 5) to manipulate a design (e.g. Fig. 10, 19) using a

plurality of heterogeneous user applications running on respective clients connected to the network, said method comprising the steps of:

connecting a session client process to a session manager over the network to participate in a collaborative session (Fig. 2, 8; col. 11 lines 44-55 – Note: Web server of hub enterprise reads on session manager – see Fig 13 - and HTTPs communication controlled by SSL and firewall reads on session for enterprise client; authentication – col 10 lines 8-67);

sharing session control messages with other session client processes connected to said session manager (Fig. 8; col. 15 lines 9-33; manager 200 - Fig. 14; col. 14 lines 43-59; Fig. 13);

loading design data representing said design into a local application running on said client (e.g. Fig 14; internal workspace, external workspace, workspace manager - col. 14 line 62 to col 15 line 5);

creating at least one application state file representing at least one application state of said local application based on at least one manipulation of said design using said local application; communicating said at least one application state file from said session client process to said other session client processes via said session manager (Fig. 13, 14);

receiving at least one application state file created by other local applications and communicated from said other session clients via said session manager (Fig. 13-14; col. 14 lines 43-59);

presenting the at least one application state file created by other local applications to a user of the session client process (*internal workspace, external workspace, workspace manager* - col. 14 line 62 to col 15 line 5);

loading the at least one application state file created by other local applications (e.g. *component-based integration, accessor, external source, passed from activity to activity* – col. 15 lines 41-65; Fig. 17, 18; col 13 lines 21-51),

Notani does not disclose “allowing the user” to delay the instantiation of the at least one application state file created by other local applications until a time determined by the user of the session client process. But the ‘allowing the user’ has been treated (see USC 112 Rejection) as though loading or activating of received versions can be time-independent of (or simultaneously coexisting with) one on-going user’s version under development within a session, and accordingly Notani teaches arrival of message or workflow data as event (event manager- Fig 12, 13) and independency of user runtime with respect to asynchronous existence of versions(col.17 lines 34-39) This entails a concurrency of actual user version within a client session and underlying arrival at the workplace manager 202; or existence other versions among the subscribers, such that each event is not dependent of the other in terms of the time when data is received and when data is integrated into user’s GUI application). It would have been obvious for one skill in the art at the time the invention was made to implement such time non-dependency so that the client in Notani can have flexibility in time whereby to decide when to use the latest received versions of a portion of communicated workflow data or object via use of Event Manager, Workflow manager and message collaborative in Notani’s Hub and Spoke system.

Nor does Notani explicitly disclose said loading such to allowing the user of the session client process to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user. But based on the interpretation set forth in the USC 112

Rejection, the “allowing the user to manipulate” limitation is interpreted as the user non-dependency between time when received versions are asynchronously available (at the Workspace manager 202) and time where the current versions (col 14 line 40 to col 15 line 67) is replaced by the incoming versions (phase in - col. 17 lines 36-39; Fig. 17-18) is not a constraint for the user. The rationale of obviousness has been addressed in the previous paragraph.

Notani does not explicitly disclose collaborative manipulating a design representing electrical or mechanical assemblies; however, Notani discloses workflow with parameter setting, and mechanical transition of material in a factory automation scheme (Fig. 10, 19; col 12 lines 20-28), where workflow being parameterized supports a industrial domain task. As it was well known at the time of invention to make, a design application can implement or support tasks in different industrial domains; and Thackston, in a distributed environment, teaches a domain application wherein collaboration is for a distributed design representing electrical or mechanical assemblies (column 1, lines 21-35; column 3, lines 55-61). It would have been obvious to one of ordinary skill in the art at the time of invention to implement the distributed collaboration on workflows of Notani with product design corresponding to workflows as found in Thackston's teaching. This implementation would have been obvious because one of ordinary skill in the art would be motivated to reduce cost and facilitate ease of development (Thackston: column 6, lines 11-15) and efficient management of complex manufacturing processes (Notani: column 1, lines 41-42).

As per claim 2, Notani discloses wherein said at least one application state is encoded using normalized XML structures to create said at least one application state file, and wherein

said at least one application state file is communicated as an XML message (column 3, line 55; and column 7, lines 47-49).

As per claim 3, Notani discloses wherein said XML structures are based on domain specific conventions defined in the context of the type of design data (column 3, lines 44-45).

As per claims 4-5, Notani discloses saving said at least one application state file in a journal file (column 14, lines 39-42; figure 14); comprising the step of scheduling said collaborative session (column 14, lines 39-42).

Notani does not explicitly disclose *recording session controls*. But Notani discloses incorporating of workflow elements into an activity based on received messages from an inter-enterprise, collaborative multi-tiered co-designers (see col. 11 line 19 to col. 12 line 21; Fig. 10) in terms of asynchronous elements imported to design and parameterize a ongoing workflow by way of “activity” instances, hence the workflow entails a sequence of actions to be recorded to support the parameterization leading to achieving the intended task (see col.13-14), a task represented by a sequence or flow for which a needed feature is retrieved via collaboration; i.e. and mapped into Notani’s workflow being instantiated. Based on the phases needed to accomplish a workflow in an Activity approach, it would have been obvious for one skill in the art at the time the invention was made to implement the session file so that this file include encoded data (as using a XML format) that would instruct type of controls or method directives to implement in sequence to achieve the workflow (recording a sequence of actions or states) because this recorded sequence would control how the feature being developed would achieve the needed steps contemplated in user’s Activity instantiated for a development session in Notani’s approach.

As per claims 6-7, Notani discloses the step of conducting a text- based conversation with said other session clients; logging in to said collaborative session and logging out of said collaborative session (column 14, lines 39-42).

As per claims 9-10, Notani discloses the step of displaying design manipulations corresponding (Fig. 17-19) to said application state file created and communicated by said other application files (col. 13, 21-51; Fig. 17-19); wherein said design is manipulated without having to transmit design images between said heterogeneous applications (using these standards – col 3 line 55; see claims 2-3 – Note: using XML protocol reads on not sending raw graphics).

As per claim 11, Notani discloses a computerized method for providing asynchronous training (Note: event manager to receive workflow data from other collaborative partners reads on asynchronous training – see Fig. 13-19) to a user to manipulate a design using a plurality of heterogeneous applications, said method comprising the steps of:

loading design data into a local application on a computer (refer to claims 1-9);
manipulating a design based on said design data using said local application (refer claims 1-9);

creating at least one application state file representing an application state of said local application based on at least one manipulation of said design using said local application (refer to claims 1-9)

saving each said at least one application state file in a journal file (refer to claims 1-4);
and

transmitting said journal file to another computer such that said at least one application state file can be loaded on said another computer and said at least one manipulation can be

reviewed using another heterogeneous application running on said another computer (refer claims 1-4),

wherein said journal file provides interactive instructions when played back on said another computer, thereby providing asynchronous training to the user (refer to claim 1-4 – Note: integration of workflow information into a user application or Activity reads on *interactive instructions when playback* – see Fig. 16-19; and *asynchronous* training because the asynchronous availability of message or application state journal is events by nature of communication protocol and event manager – see Fig. 13-14).

Notani does not disclose design representing electrical or mechanical assemblies; but this has been addressed in claim 1.

As per claim 12, Notani discloses wherein said method is an asynchronous method of collaboration.

As per claim 17, Notani discloses a computer program product comprising a storage medium having computer-readable code stored thereon for collaborating over a network to manipulate a design using a plurality of heterogeneous applications running on clients connected to the network, said system comprising:

at least one application state file for representing application states created by a local application (refer to claim 1; Fig. 13-14) based on at least one manipulation of a design made using said local application (Fig. 16-19); and

a session client module for synchronizing interaction between said local application and a collaborative session and for communicating said application state file over the network to other clients(refer to claim 1) and

receiving at least one other application state file from the other clients, presenting the at least one other application state file to a user (refer to claim 1);

Notani does not disclose design representing electrical or mechanical assemblies; but this has been addressed in claim 1.

Nor Notani explicitly disclose allowing the user to delay the instantiation of the at least one other application state file until a time determined by the user of the session client module, and loading the at least one other application state file, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user.

But this limitation has been interpreted (see USC 112 Rejection) and accordingly addressed in claim 1.

As per claim 18, Notani discloses wherein said session client module comprises:
a user interface for interfacing with a user;
a first application programming interface for interfacing with said local application (Activity - Fig. 17-18); and
a second application programming interface for interfacing with a session manager over the network (Manager 200, 202 - Fig. 13-14).

As per claims 19-20, Notani discloses a session journal for recording said application states during said collaboration session, wherein said application states are encoded using normalized XML structures to create said at least one application state file, and wherein said at least one application state file is communicated as an XML message (refer to claims 2-3).

Notani does not explicitly disclose recording *session controls*. But this has been addressed in claim 4.

As per claim 21, Notani discloses wherein said session client module communicates said application state file without sending images of said design (refer to claim 10).

As per claim 22, Notani discloses a computer program product comprising a storage medium having computer-readable code stored thereon for providing asynchronous training to a user to manipulate a design using a plurality of heterogeneous applications, said system comprising means for:

loading design data into a local application on a computer;

manipulating a design based on said design data using said local application;

creating at least one application state file representing an application state of said local application based on at least one manipulation of said design using said local application;

saving each said at least one application state file in a journal file; and

transmitting said journal file to another computer such that said at least one application state file can be loaded on said another computer and said at least one manipulation can be reviewed using another heterogeneous application running on said another computer,

wherein said journal file provides interactive instructions when played back on said another computer, thereby providing asynchronous training to the user;

all of which having been addressed in claim 11.

Notani does not explicitly disclose design representing electrical or mechanical assemblies; but this has been addressed in claim 1.

As per claim 23, Notani discloses a computer program product comprising a storage medium having computer-readable code stored thereon for collaborating over a network to manipulate a design using a plurality of heterogeneous applications on client computers connected to the network, said computer-readable code comprising:

code for interfacing with a user; code for interfacing with a local application used to manipulate a design (refer to claim 18);

code for interfacing with a session manager over a network (refer to claim 1);

code for creating a session client process to communicate session controls and at least one application state file to said session manager over said network (refer to claim 1), wherein said application state file is created by said local application based on at least one manipulation of said design (refer to claim 1); and

code for receiving at least one other application state file from said session manager over said network, presenting the at least one other application state file to the user (refer to claim 1)

Notani does not explicitly disclose design representing electrical or mechanical assemblies; but this has been addressed in claim 1.

Nor Notani explicitly disclose allowing the user to delay the instantiation of the at least one other application state file until a time determined by the user of the session client module, and loading the at least one other application state file, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user.

But this limitation has been interpreted (see USC 112 Rejection) and accordingly addressed in claim 1

As per claim 24, Notani discloses a computer program product comprising a storage medium having computer-readable code stored thereon for collaborating over a network to manipulate a design using a plurality of heterogeneous applications on client computers connected to the network, said computer-readable code comprising:

code for interfacing with a user; code for interfacing with a local application;
code for instructing said local application to create at least one application state file representing at least one application state based on a manipulation made to a design using said application state file;
code for interfacing with a session manager over a network;
code for sending said at least one application state file and session controls to said session manager;

all of which addressed in claim 1.

code for notifying said local application that at least one other application state file has been received, presenting the at least one other application state file to the user (Fig. 12-13 and related text),

Notani does not explicitly disclose design representing electrical or mechanical assemblies; but this has been addressed in claim 1.

Nor Notani explicitly disclose allowing the user to delay the instantiation of the at least one other application state file until a time determined by the user of the session client module, and loading the at least one other application state file, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by

another user. But this limitation has been interpreted (see USC 112 Rejection) and accordingly addressed in claim 1

As per claim 25, Notani discloses code for controlling instantiation of said received application state file (Fig. 16-19; col 13 lines 21-51) into said local application.

As per claim 26, Notani discloses a computerized method for collaborating over a network to manipulate a design using a plurality of heterogeneous user applications running on respective clients connected to the network, said method comprising the steps of:

connecting a session client process to a session manager over the network to participate in a collaborative session(refer to claim 1);

sharing session control messages with other session client processes connected to said session manager (re claim 1; Fig. 6);

loading design data representing said design into a local application running on said client (e.g. Fig. 10-19);

creating at least one local application state event representing at least one application state of said local application based on at least one manipulation of said design using said local application (refer to claim 1); and

communicating said at least one local application state event from said session client process to said other session client processes via said session manager (refer to *application state file* in claim 1);

receiving at least one other application state file from said other session client, presenting the at least one other application state file to a user of the session client process (refer to claim 1).

Notani does not explicitly disclose design representing electrical or mechanical assemblies; but this has been addressed in claim 1.

Nor Notani explicitly disclose allowing the user to delay the instantiation of the at least one other application state file until a time determined by the user of the session client module, and loading the at least one other application state file, thereby allowing the user to manipulate a first aspect of the design before loading changes made to a second aspect of the design by another user. But this limitation has been interpreted (see USC 112 Rejection) and accordingly addressed in claim 1

As per claim 27, Notani discloses communicating further comprises communicating said at least one local application state event as at least one data packet (TCP/IP – col 11 lines 36-55) for representing said at least one application state event.

As per claim 28, Notani discloses wherein the at least one local application state event is at least one of a plurality of normalized application state events recognized by each of the heterogeneous user applications (XML reads on normalized data being formatted as message – refer to claims 2-3).

As per claim 29, Notani discloses saving said session control messages and said at least one local application state event in a journal file (refer to claim 10).

As per claim 30, Notani discloses user interface (Fig. 10-19) with integration of received collaboration data (col. 13 lines 21-51) hence the desirability of accept/refuse is given to the user by virtue of well-known concepts. Hence option given for the session user to refuse (see USC 112 Rejection) the instantiation of the at least one application state file created by other local applications would have been obvious, because otherwise the user application would be

subject to import of unauthorized data into its environment leading to conflicts or undesired effects.

As per claim 31, Notani discloses buffering the at least one application state file created by other local applications (column 5, lines 35-47, the workspace itself, column 5, lines 51-53, buffered by user control).

Response to Arguments

6. Applicant's arguments filed 3/30/09 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

USC § 103(a) Rejection:

- (A) Applicants have submitted that the cited portion (Fig. 5 by Notani) does not teach or suggest 'presentation of application state files' from other clients and 'allowing the user to delay ... the instantiation' (Appl. Rmrks pg. 12 top). The current rejection has been necessitated to address the newly added limitation 'allowing the user to delay'. The argument is largely moot.
- (B) Applicants have submitted that Thackston and Notani cannot teach or disclose 'allows the user to delay' and 'to work on a first aspect of the electrical or mechanical assembly before loading' (Appl. Rmrks pg. 12) The added limitations have been addressed correspondingly in the new grounds of rejections. The argument is moot.
- (C) For claims 11, 12, 22 Applicants have submitted that Notani in view of Thackston cannot teach or suggest 'asynchronous training' by transmitting a journal file as claimed (Appl. Rmrks pg. 13) and that Office Notice was improperly applied. The rejection has addressed 'journal' in terms of broad interpretation (i.e. some message/record that is published and communicated to a group inside the collaborative system), and has addressed what can be analogized to

‘asynchronous training’. Mere allegation that the references do not teach cannot properly establish how the language of the claim (i.e. without details about how the claimed feature is implemented) would obviate the use of the reference, i.e. by pointing out precisely how the cited parts distinguish from the claimed feature. The argument is therefore non compliant to the very details of the rejection (e.g. the term ‘Official Notice’ being nowhere mentioned). Applicant’s arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

(D) Applicant’s plead that the invention is a non-obvious advantage over prior arts such as Notani and Thackston remains a non-factual case of rebut, notwithstanding the deficiency of the claimed language in terms of USC 112 statutory category.

In all, the claims stand rejected as set forth in the Office Action.

Conclusion

7. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu /

Primary Examiner, Art Unit 2193

May 25, 2009